COMPLETE PROOF RULES FOR STRONG FAIRNESS AND STRONG EXTREME-FAIRNESS†

Michael G. Main

CU-CS-447-89 August 1989
Corrections added February 1990

† This research has been supported in part by National Science Foundation grant CCR-8701946,

COMPLETE PROOF RULES FOR STRONG FAIRNESS AND STRONG EXTREME-FAIRNESS†
August 1989
Corrections added February 1990

Michael G. Main
Department of Computer Science
University of Colorado
Boulder, CO 80309 USA

ABSTRACT

This paper demonstrates completeness of a termination-rule for iterative programs with strongly fair nondeterminism, even when there are countably infinite options for the nondeterminism. This means that whenever a program is guaranteed to terminate under the assumption of strong fairness, then this termination can be proved via the strongly fair termination rule. A variant of the rule is also shown to be complete for extremely-fair nondeterminism, as introduced by Pnueli [9] and developed by Francez [6, Section 4.3].

† This research has been supported in part by National Science Foundation grant CCR-8701946,

1. Introduction

In this paper, we consider repetition statements of the following form, where each b_i is a Boolean expression (also called a *guard*), and each s_i is an executable command.

Repetition Statement	Informal Meaning
*[$b_1 \rightarrow s_1$ □ $b_2 \rightarrow s_2$ □ ...]	1. If some expression b_i is true, then select such an i and execute the corresponding command s_i. Then repeat step 1. 2. Otherwise, stop.

A repetition statement is nondeterministic, because sometimes there are several true guards, only one of which is selected for execution — and the selection mechanism is unspecified.

This paper shows completeness of a proof rule for showing termination of this kind of repetition statement, under various "fairness" assumptions about the nondeterministic selection mechanism. This means that whenever a repetition statement is guaranteed to terminate under the fairness assumptions, then this termination can be proved via the proof rule. This completeness result is new because it includes statements with countably infinite directions, and also applies to "extreme fairness" as introduced by Pnueli [9] and developed by Francez [6, Section 4.3].

The presentation in this paper assumes a familiarity with well-founded sets. The other background for the results is contained in Section 2 of this paper, although a knowledge of the first four chapters of Francez [6] would also be helpful.

2. Strongly Fair Termination and the SFT Proof Rule

2.1 Strong Fairness

When a repetition statement selects between several true guards, the selection is completely arbitrary. For example, consider this repetition statement:

```
*[
      x > 0 → y := y+1
      □
      x > 0 → x := 0
]
```

Suppose that x starts with a non-zero value. Then it is possible to always select the first option, continually incrementing y and never terminating. However, a *strongly fair* execution path would not permit this: an execution path is *strongly fair* provided that whenever a guard is true infinitely often, then the corresponding statement is selected infinitely often. For the above example, there are no nonterminating strongly-fair execution paths, since any strongly fair path must eventually select the second option, causing termination.

As a second example, consider this repetition statement:

```
*[
      x > 0 → y := y+1
      □
      x > 0 → x := x−1
]
```

There are nonterminating execution paths: those paths which choose the second option fewer than n times, where n is the initial value of x. However, under the assumption of strong fairness, termination is guaranteed — since any infinite strongly-fair path must select the second option infinitely often, eventually making $x > 0$ fail.

2.2 Countably Infinite Directions

Dijkstra [4] introduced the repetition statement as a convenient way of presenting nondeterministic alternatives, and the notation has since been used in other studies of nondeterminism, concurrency and fairness. But usually the notation is restricted to finitely many options. The restriction is not an oversight, but rather a decision by Dijkstra to include a ''Law of Continuity'' which fails for countably infinite nondeterminism (see [5, Chapter 9]).

The Law of Continuity also fails when we restrict ourselves to strongly fair execution paths, and there seems no further harm in allowing countably many options in a repetition statement. Informally, we might have a statement like this:

```
*[
      x ≠ 0  →  x := abs(y) − 0
      □
      x ≠ 0  →  x := abs(y) − 1      (Note: abs(y) is the absolute value of y.)
      □
      x ≠ 0  →  x := abs(y) − 2
      ...
]
```

This statement has nonterminating execution paths — simply avoid the one option that will cause x to be set to zero. But there are no infinite strongly-fair execution paths, since any infinite strongly-fair path must eventually select that one option which sets x to zero, causing termination.

Here's another example of a statement which has no infinite strongly-fair execution paths:

```
*[
      x > 0  →  x := x+1;  y := maximum(x, y)
      □
      x > 0  →  x := x−1
      □
      x > 0 and x = y  →  x := 0
      □
      x = 1  →  x := 0
      □
      x = 2  →  x := 0
      □
      x = 3  →  x := 0
      ...
]
```

An infinite path may only select from the first two options. Such a path has two possibilities: Either x remains below some fixed bound all the time, or there is no such fixed bound. If there is no such fixed bound, then $x = y$ will hold infinitely often, and the path is not strongly fair, because the third option was never selected. On the other hand, if there is such a bound, then there exists some k below the bound such that $x = k$ is true infinitely often. Again, the path is not strongly fair, because the option with guard $x = k$ is never selected. This shows that there is no infinite strongly-fair path.

The topic of this paper is a proof rule for showing this kind of termination result. But first we need to formalize the notions of repetition statement and execution paths.

2.3 Repetition Statements and Execution Paths

Our formal definition of a repetition statement is with respect to a fixed set of ''computation states''. Throughout the rest of the paper, Σ will be this fixed set of states. We use the term *predicate* for any Boolean-valued function on Σ, and we write $\neg p$ for the negation of a predicate p. For a set Γ of predicates, we write $\bigvee\Gamma$ for the disjunction of all the predicates in Γ. We use $\Sigma_\perp$ to denote the set $\Sigma \cup \{\perp\}$, where $\perp$ is a new element which represents nontermination of a process.

A nondeterministic program over Σ can be interpreted as a state-transition relation between Σ and $\Sigma_\perp$. If s is a relation denoting such a program, and $(x, y) \in s$, then the corresponding program is capable of mapping an initial state x to a final state y. If $y = \perp$ then the program has a nonterminating execution path starting in state x. Such a relation should also be *total*, so that for each $d \in \Sigma$ there exists at least one $e \in \Sigma_\perp$ such that d is related to e.

Now we can define a repetition statement:

Definition. (Repetition Statement) Let $<b_1, s_1>$, $<b_2, s_2>$ ... be a countable sequence of ordered pairs, where each b_i is a predicate, and each s_i is a total relation from Σ to $\Sigma_\perp$. Such a sequence is a *repetition statement* and usually written as:

```
*[
      b1 → s1
      □
      b2 → s2
      □
      ...
]
```

If R is a repetition statement, then the *directions* of R are the natural numbers 1,2,..., up to the number of pairs in the sequence (or unbounded if the sequence is infinite). □

Issues of syntax and computability are intentionally vague in this definition. These issues could be clarified, but that would complicate some proofs without changing the results. In particular, we could require that each command s_i is a repetition statement or basic command of some sort — the results of the paper would still hold, but most proofs would need an extra induction on the nesting level of the program (which would need to be restricted to be finite).

We associate a set of execution paths with each repetition statement, as follows:

Definition. (Execution Path) Let R be a repetition statement with guards $b_1, b_2, \cdots$ and executable statements $s_1, s_2, \cdots$. A *finite execution path for R* is a finite sequence $x_1, d_1, x_2, d_2, \cdots, x_k$, such that

a. Each x_i is a state from $\Sigma_\perp$;

b. Each d_i is a direction for the statement R;

c. For all i (with d_i defined): b_{d_i} holds in state x_i;

d. For all i (with d_i defined): x_i is related to x_{i+1} by the relation s_{d_i};

e. Either $x_k = \perp$, or none of the guards of R holds for x_k.

An *infinite execution path for R* is a countably infinite sequence $x_1, d_1, x_2, d_2, \cdots$, meeting conditions (a)-(d). □

Intuitively, an execution path for R is a sequence of states and direction choices which the statement R could pass through. Note that $\perp$ can only appear as the last state of a finite execution path. The intuition behind a $\perp$-ending path is that the last executable statement did not terminate. As described above, some paths are strongly fair. Here's the formal definition of strongly fair, along with two other definitions that we'll use:

Definition. (Strongly Fair Execution Path) Let R be a repetition statement, as in the previous definition. Let $x_1, d_1, x_2, d_2, \cdots$ be a (possibly infinite) execution path for R. The path is *strongly fair* provided that the following holds for every direction d of R:

If $\{i \mid b_d$ holds for $x_i\}$ is infinite, then direction d appears infinitely often in the sequence.

□

Definition. (Total Correctness Notation) Let p and q be predicates, and let s be a total relation from Σ to $\Sigma_\perp$. We say that s is *totally correct* with respect to precondition p and postcondition q provided that whenever p holds for a state x and $(x,y) \in s$ then $y \neq \perp$ and q holds for state y. In this case, we write $\{p\}\, s\, \{q\}$. □

Definition. (Strongly Fair Termination) For a repetition statement R and a predicate p, we use the notation SFT(p,R) to denote the statement that R has no strongly-fair execution path that starts in a state that satisfies p and is infinite or ends with $\perp$. □

2.4 Proof Rule for Strongly Fair Termination

Our goal is to prove statements of the form SFT(p,R). In other words, if R is executed in an environment that guarantees strong fairness, and it starts in a state that satisfies p, then R is certain to terminate. Figure 1 shows one form of the established proof rule for proving these statements. The rule is based on Francez [6, Page 40], whose original source was research by Grumberg, Francez, Makowsky and deRoever [7]. Similar methods were also proposed by others [1,2,3,8].

Let p be a predicate, and let R be a repetition statement:

*[
    $b_1 \rightarrow s_1$
    □
    $b_2 \rightarrow s_2$
    □
    ...
]

To prove SFT(p, R): Choose a well-founded, partially-ordered set $(W, <)$, a predicate $pi(w)$ for each $w \in W$, and a direction d_w for each non-minimal $w \in W$, all satisfying:

1. (INIT) p implies there exists w such that $pi(w)$.
2. (TERM) For all minimal $w \in W$: $pi(w)$ implies none of R's guards hold.
3. (DEC) For all non-minimal $w \in W$:

    $\{pi(w)$ and $b_{d_w}\}$ s_{d_w} {There exists v such that $v < w$ and $pi(v)\}$.

4. (NOINC) For all non-minimal $w \in W$ and all directions i:

    $\{pi(w)$ and $b_i\}$ s_i {There exists v such that $v \leq w$ and $pi(v)\}$.

5. (IOE) For all non-minimal $w \in W$, SFT($pi(w), \bar{R}$) can be proved by an application of this rule, where $\bar{R}$ is the following repetition statement:

*[
    $(\neg b_{d_w})$ and $b_1 \rightarrow s_1$
    □
    $(\neg b_{d_w})$ and $b_2 \rightarrow s_2$
    □
    ...
]

Figure 1. SFT Rule

The names INIT, TERM, etc. (see Figure 1) are from Francez's text, and stand for ''initialization'', ''termination'', ''decrement'', ''no increase'', and ''infinitely often enabled''. The predicate $pi(w)$ is called the ''parameterized invariant for w'', and the direction d_w is called the ''helpful direction for w''. This SFT rule is slightly simpler than Francez's rule, since Francez's rule forbids termination when $pi(w)$ holds for a non-minimal w. Also, the rule in Figure 1 has only one helpful direction for each w (instead of a set of helpful directions). But the rule remains sound (as shown below) and complete (as shown in the Section 5).

We write $\vdash \mathrm{SFT}(p, R)$ when it is possible to prove $\mathrm{SFT}(p, R)$ with the SFT rule of Figure 1. The next definition shows precisely what is meant by ''provable''.

Definition. Here is a recursive definition of when $\mathrm{SFT}(p, R)$ is provable for a repetition statement R and a predicate p

(1) There are some instances of $\mathrm{SFT}(p, R)$ which can be proved with the SFT rule and no recursive applications needed by the IOE condition. These are the cases where W has only minimal elements. Whenever this is the case, then $\mathrm{SFT}(p, R)$ is provable.

(2) Let K be any set of provable statements of the form $\mathrm{SFT}(q, S)$, and suppose $\mathrm{SFT}(p, R)$ follows from the SFT rule, where each termination statement needed by IOE occurs in K. Then $\mathrm{SFT}(p, R)$ is provable.

(3) The statement $\mathrm{SFT}(p, R)$ is not provable unless this is required by rule 1 or rule 2. (Note that rule 1 is actually a special case of rule 2, where K is empty.)

Whenever $\mathrm{SFT}(p, R)$ is provable, we write $\vdash \mathrm{SFT}(p, R)$. □

In other words, the set of provable statements is the smallest set of statements which is consistent with the proof rule. To demonstrate that the SFT rule is sound and complete, we must show that for any predicate p and repetition statement R, $\vdash \mathrm{SFT}(p, R)$ if and only if $\mathrm{SFT}(p, R)$. One direction of this (soundness) is pretty easy and the usual soundness proof for SFT (*e.g.*, [6, page 44]) works even with countably

infinite directions. That proof is given here:

Theorem: (Soundness of SFT) For a repetition statement R and a predicate p:

$$\text{if } \vdash \text{SFT}(p, R), \text{ then } \text{SFT}(p, R).$$

Proof: The proof is an induction on the recursive definition of $\vdash \text{SFT}(p, R)$. To set up the induction, assume that $\vdash \text{SFT}(p, R)$ holds, so that there is a set K of provable statements of the form $\text{SFT}(q, S)$, and suppose $\text{SFT}(p, R)$ follows from the SFT rule, where each termination statement needed by IOE occurs in K (which might be empty). For the induction hypothesis, we assume that whenever $\text{SFT}(q, S)$ is in K, then $\text{SFT}(q, S)$ is actually valid. We must show that $\text{SFT}(p, R)$ is also valid. First we note that R has no finite execution path that starts in a state that satisfies p and ends with $\perp$. (This follows from INIT and NOINC). Next, consider some path $\pi = x_1, d_1, x_2, d_2, \cdots$, which is infinite execution path of R, such that p holds for state x_1. We must show that π is not strongly fair. By INIT and NOINC, there exists a sequence of elements of W, $w_1 \geq w_2 \geq w_3 \cdots$, such that for every j, $pi(w_j)$ holds for state x_j. By well-foundedness, we may choose the w_j so that there is never some $v < w_j$ where $pi(v)$ holds for x_j. Also from well-foundedness, there exists some $k > 0$, such that $w_k = w_{k+1} = \cdots$. But this implies that none of the directions d_k, d_{k+1}, $\cdots$ are d_{w_k} (since DEC indicates that after a d_{w_k} direction is taken, there will be some $v < w_k$ such that $pi(v)$ holds). However, by IOE (and the induction hypothesis), the guard for direction d_{w_k} is infinitely-often true in the states x_k, x_{k+1}, $\cdots$. Since direction d_{w_k} is never taken from these states, the path π is not strongly fair. □

3. Examples and Properties of the Strongly Fair Termination Rule

Francez [6] provides many examples of applications of the SFT rule. Here are a few more.

Example. Consider the repetition statement, which we will call R in this paragraph:

```
*[
      x > 0  →  y :=y+1
      □
      x > 0  →  x :=x−1
 ]
```

In order to show $\vdash \mathrm{SFT}(\mathit{true}, R)$, we can take the well-founded set W to be the natural numbers, with the usual ordering. For any natural number n, we define the parameterized invariant $pi(n)$ to be the predicate $x \leq n$. And for any $n > 0$, we choose the helpful direction d_n to be the second direction. It is not difficult to show that the conditions of the SFT rule are valid for these choices. Note that for each $n > 0$, IOE requires another application of the SFT rule to show $\mathrm{SFT}(x \geq n, \bar{R})$, where $\bar{R}$ is the same as R, except the guards are now $[(\neg(x > 0))$ and $(x > 0)]$. Since all these guards are equivalent to "false", this is an easy application of the SFT-rule (see Lemma 3.1, below).

Example. Consider the repetition statement, which we will call R in this paragraph:

```
*[
      x > 0 → x := 0
      □
      x > 0 → x := 1
      □
      x > 0 → x := 2
      □
      x > 0 → x := 3
      ...
]
```

In order to show $\vdash \mathrm{SFT}(\mathit{true}, R)$, we can take the well-founded set W to be the two element set $\{\mathit{top}, \mathit{bottom}\}$ with $\mathit{top} > \mathit{bottom}$. We also define these:

$$pi(\mathit{top}) = x > 0,$$

$$pi(\mathit{bottom}) = x \leq 0,$$

$$d_{\mathit{top}} = \text{``the first direction''}.$$

It's easy to see that the conditions INIT, TERM, NOINC and DEC are all satisfied. For IOE we need to show $\vdash \mathrm{SFT}(x > 0, \bar{R})$, where $\bar{R}$ is obtained by adding $\neg(x > 0)$ to each guard of R. But this makes all the guards equivalent to "false", so this is another easy application of the SFT-rule (see Lemma 3.1, below).

Example. Let m be a constant natural number, and consider this repetition statement, which we call R_m:

```
*[
      x ≥ y − m and x ≠ y and x > 0 → x := x + 1; y := maximum(x, y)
      □
      x ≥ y − m and x ≠ y and x > 0 → x := x − 1
      □
      x ≥ y − m and x ≠ y and x = 1 → x := 0
      □
      x ≥ y − m and x ≠ y and x = 2 → x := 0
      □
      x ≥ y − m and x ≠ y and x = 3 → x := 0
      ...
]
```

Let m and n be natural numbers with $m < n$. We can show $\vdash \mathrm{SFT}(y - m \leq x < y = n, R_m)$, by induction on the value of m. For the base case

(when $m=0$), the condition $y-m \le x < y = n$ is equivalent to "false", and $\vdash \mathrm{SFT}(false, R_m)$ is immediate from Lemma 3.1 (below). For the induction step, assume that $\vdash \mathrm{SFT}(y-k \le x < y = n, R_k)$ holds for some value k. From this assumption, we need to show that $\vdash \mathrm{SFT}(y-(k+1) \le x < y = n, R_{k+1})$ also holds. To do this, we define a well-founded set W to contain two elements $\{top, bottom\}$ with $top > bottom$. We also define these:

$$pi(top) = y-(k+1) \le x < y = n,$$

$$pi(bottom) = x \le y-(k+1) \text{ or } x = y,$$

and d_{top} = the direction whose guard is "$\cdots$ and $x = y-k$".

It's easy to see that the conditions INIT, TERM, NOINC and DEC are all satisfied. For IOE we need to show $\vdash \mathrm{SFT}(y-(k+1) \le x < y = n, \bar{R})$, where $\bar{R}$ is obtained by adding $\neg d_{top}$ to each guard of R_{k+1}. But, adding $\neg d_{top}$ to each guard of R_{k+1} gives the statement R_k, so we need only show:

$$\vdash \mathrm{SFT}(y-(k+1) \le x < y = n, R_k).$$

And this follows from the induction hypothesis ($\vdash \mathrm{SFT}(y-k \le x < y = n, R_k)$) and Lemma 3.6 (which is given below and allows us to strengthen the precondition from $y-k \le x$ to $y-(k+1) \le x$).

Exercise. Show $\vdash \mathrm{SFT}(true, R)$, where R is this repetition statement, from Section 2.2:

```
*[
      x > 0  →  x := x+1;  y := maximum(x,y)
      □
      x > 0  →  x := x-1
      □
      x > 0 and x = y  →  x := 0
      □
      x = 1  →  x := 0
      □
      x = 2  →  x := 0
      □
      x = 3  →  x := 0
      ...
]
```

The previous example will be useful in the recursive applications of SFT that are required by IOE.

It is useful to have some general results that indicate when $\vdash \mathrm{SFT}(p, R)$ holds. These are given in the rest of this section. Throughout these results, R is a repetition statement, with guards $b_1, b_2, \cdots$ and executable statements $s_1, s_2, \cdots$.

Lemma 3.1. Let p be a predicate such that p implies $\neg b$ for each guard b of R. Then $\vdash \mathrm{SFT}(p, R)$.

Proof: Let the well-founded set W be the one-point set, with p as the parameterized invariant at this one point. □

Lemma 3.2. Let p be a predicate such that $\vdash \mathrm{SFT}(p, R)$, and let R' be the same as R, but with stronger guards — so that each guard in R is implied by the corresponding stronger guard in R'. Then $\vdash \mathrm{SFT}(p, R')$.

Proof: We can use the same well-founded set for R' as for R, with the same parameterized invariants and the same helpful directions. □

Lemma 3.3 Let Γ be a set of predicates such that for every $p \in \Gamma$: $\vdash \mathrm{SFT}(p, R)$. Then $\vdash \mathrm{SFT}(\vee \Gamma, R)$.

Proof: Create the well-founded set for $\mathrm{SFT}(\vee \Gamma, R)$ as the disjoint union of the well-founded sets for all of the individual $p \in \Gamma$. This new well-founded set inherits its parameterized invariants and helpful directions from the original well-founded sets for the individual $p \in \Gamma$. □

Lemma 3.4 Let p and q be predicates such that $\vdash \mathrm{SFT}(q, R)$ and for every direction i of R:

$$\{p \text{ and } b_i\}\ s_i\ \{q \text{ or none of } R\text{'s guards hold}\}.$$

Then $\vdash \mathrm{SFT}(p, R)$.

Proof: Start with the well-founded set for proving $\mathrm{SFT}(q, R)$. For each direction i add one new element w_i at the top (so that $w_i > v$ for every v in the original well-

founded set). We also add a new element w which is not related to any other element, and we define:

$$\text{For all } i,\ pi(w_i) = p \text{ and } b_i,$$

$$\text{For all } i,\ d_{w_i} = i,$$

$$pi(w) = p \text{ and none of } R\text{'s guards hold.}$$

It's not difficult to show that the conditions of the SFT rule are still valid for this well-founded set. □

Lemma 3.5 Let p be a predicate and Γ be a set of predicates such that for every direction i there exists some $q \in \Gamma$ such that $\vdash \mathrm{SFT}(q, R)$ and

$$\{p \text{ and } b_i\}\ s_i\ \{q \text{ or none of } R\text{'s guards hold}\}.$$

Then $\vdash \mathrm{SFT}(p, R)$.

Proof: The result follows from the previous two lemmas. □

Lemma 3.6 Let p and q be predicates such that $\vdash \mathrm{SFT}(q, R)$ and p implies q. Then $\vdash \mathrm{SFT}(p, R)$.

Proof: We can use the same well-founded set for p as for q, with the same parameterized invariants and the same helpful directions. □

4. Completeness of the Strongly Fair Termination Rule

Throughout this section, p is a predicate and R is a repetition statement such that $\mathrm{SFT}(p, R)$. As usual, R has guards $b_1, b_2, \cdots$ and executable statements $s_1, s_2, \cdots$.

The main result of this section is that the SFT rule is always adequate for proving strongly fair termination. Thus, we will show that our assumption of $\mathrm{SFT}(p, R)$ implies $\vdash \mathrm{SFT}(p, R)$. Here's an outline of the proof technique:

(1) From p and R, we construct a well-founded set G. Each element $g \in G$ has a statement $\mathrm{SFT}(\mathit{invariant}(g), R_g)$ associated with it.

(2) We use well-founded induction on G to prove that $\vdash \mathrm{SFT}(\mathit{invariant}(g), R_g)$ holds for every $g \in G$. For an arbitrary $g \in G$, this well-founded induction assumes (as the induction hypothesis) that $\vdash \mathrm{SFT}(\mathit{invariant}(w), R_w)$ holds for any $w < g$. From this induction hypothesis, we directly show $\vdash \mathrm{SFT}(\mathit{invariant}(g), R_g)$, by constructing a well-founded set W (together with a parameterized invariant and helpful directions). The well-founded set meets the requirements of the SFT rule — in particular, it meets the requirments of IOE, which have the form $\vdash \mathrm{SFT}(\cdots)$. This demonstration of IOE uses the induction hypothesis. In some instances, the statement $\vdash \mathrm{SFT}(\cdots)$ which is needed for IOE may be no simpler than $\vdash \mathrm{SFT}(\mathit{invariant}(g), R_g)$ — in fact it may even be identical! This just means that a statement similar (or even identical) to $\vdash \mathrm{SFT}(\mathit{invariant}(g), R_g)$ was proved at a lower point in the well-founded set G. From the induction hypothesis we can make use of that similar (or identical) statement.

(3) After we have shown $\vdash \mathrm{SFT}(\mathit{invariant}(g), R_g)$ holds for all $g \in G$, we show that this implies $\vdash \mathrm{SFT}(p, R)$.

This proof technique is not as straight-forward as the completeness proof for a finite number of directions. In the finite case each recursive application of SFT is simpler than the previous one — because one direction of the repetition statement has been removed. In the infinite-directions case, the only thing getting simpler in the recursive applications is that such an application occurs lower in the well-founded set G.

The proof of $\vdash \mathrm{SFT}(p, R)$ uses some definitions, which are given in the next three paragraphs.

Definition 4.1. Let $e_0, e_1, e_2, \cdots$ be some fixed infinite sequence of directions from the repetition statement R, such that every direction of R appears infintely often. □

Definition 4.2. Let $x \in \Sigma$ be a state and q be a predicate. Then $\mathit{reachable}(x, q)$ is the set of states that can be reached (in an execution of R) starting at x and never passing through a state where q holds. Thus, $y \in \mathit{reachable}(x, q)$ if and only if:

For some k and some execution path $x_1, d_1, \cdots, d_{k-1}, x_k, \cdots$:

$x = x_1$, and

$y = x_k$, and

q fails for all of the states $x_1, \cdots, x_k$.

Note: If q holds in state x, then $reachable(x, q)$ is empty. If q fails in state x, then $reachable(x, q)$ always contains at least x. □

Definition 4.3. Let j be some direction of R, and let q be a predicate. Then $step(j, q)$ is the set of states which can be reached by executing s_j from a state where q and b_j hold. Formally, it is the set

$$\{y \in \Sigma \mid \text{There exists } x \in \Sigma \text{ such that } q(x) \text{ and } b_j(x) \text{ and } (x, y) \in s_j\}$$

□

The next definition gives a partially-ordered set $(G, <)$, where each element $g \in G$ is labeled by two predicates (called $invariant(g)$ and $forbidden(g)$) and one state (called $state(g)$) and one integer (called $level(g)$).

Some intuition might help in understanding the definition: Consider the possibility of an infinite strongly-fair execution path π of R which begins in some state x which satisfies p. G will be constructed so that there is some $g \in G$ with $level(g) = 0$ and $state(g) = x$. For this g, $invariant(g)$ will be the set of states that can be reached from x along some execution path, and $forbidden(g)$ will be empty.

Now, consider direction e_0 (from the sequence in Definition 4.1). Since the path π is strongly-fair, there are two possibilities:

(1) Direction e_0 is choosen somewhere in the path. In this case, there will be some $h \in G$ where $level(h) = 1$ and $state(h)$ is the state of the path after taking direction e_0. The set $forbidden(h)$ will still be empty, and $invariant(h)$ will be the set of states that can be reached from $state(h)$ along some execution path.

(2) Eventually there is some point in the path where the guard b_{e_0} is never again satisfied. Direction e_0 is never taken in the path. Since the path is strongly-

fair, there must be some point in the path where the guard b_{e_0} is never again satisfied. In this case, there will be some $h \in G$ where $level(h) = 1$ and $state(h)$ is the state of the path after reaching this point. The set $forbidden(h)$ now includes any state that satisfies b_{e_0} — so intuitively, these states have been "forbidden" to occur on the path in the future. The set $invariant(h)$ will include all states that can be reached from $state(h)$ without passing through one of the forbidden states.

As an execution path proceeds, we will be able to follow it through higher and higher levels of G. When the computation proceeds from level i to level $i+1$, the choice of the element of G depends on whether direction e_i is ever taken again.

One more bit of intuition before the actual definition of G: From the fact that there are no infinite strongly-fair execution paths, we will show that G is well-founded. And from the well-foundedness of G we will show $\vdash \mathrm{SFT}(p, R)$.

Definition 4.4. The elements of G depend on R, p, and the sequence $e_0, e_0, e_2, \cdots$. They are recursively defined as follows:

(a) For each state x such that p holds, there is an element g of G such that:

$forbidden(g) = false$,

$invariant(g) = reachable(x, false)$,

$state(g) = x$, and

$level(g) = 0$.

Note that $invariant(g)$ contains at least x.

(b) Let $g \in G$, let $i = level(g)$, let j be a direction and let x be a state in $step(j, invariant(g))$. Then there is an element $h \in G$ with $h < g$ such that:

$forbidden(h) = forbidden(g)$ or b_{e_i},

$invariant(h) = reachable(x, forbidden(h))$,

$state(h) = x$, and

$level(h) = i+1$.

(c) Let $g \in G$, let $i = level(g)$, and let $x \in step(e_i, invariant(g))$. Then there is an element $h \in G$ with $h < g$ such that:

$forbidden(h) = forbidden(g)$,
$invariant(h) = reachable(x, forbidden(h))$,
$state(h) = x$, and
$level(h) = i+1$.

There are no elements or relations in G, except those required by the above three rules. □

Eventually we will use G to prove $\vdash SFT(p, R)$. The first step toward this is to show that G is well-founded:

Lemma 4.5. The partially-ordered set $(G, <)$ is well-founded.

Proof: For the sake of reaching a contradiction, assume there is an infinite sequence $g_0 > g_1 > g_2 \cdots$ of elements in G. This sequence can be extended before g_0 to a level 0 element of G, so without loss of generality, we can assume that g_0 is at level 0, and that each g_k $(k \geq 0)$ is at level k. From the definition of G, it is straightforward to construct an infinite strongly-fair execution path that starts in g_0, and continues passing through states $g_1, g_2, \cdots$, in such a way that for all k, none of the states at or after $state(g_k)$ satisfies $forbidden(g_k)$. This contradicts $SFT(p, R)$, and by this contradiction, $(G, <)$ is well-founded. (End of proof of Lemma 4.5.) □

Since $(G, <)$ is well-founded, we can provide an induction proof of the next lemma:

Lemma 4.6. For each $g \in G$: $\vdash SFT(invariant(g), R_g)$, where R_g is the repetition statement:

```
*[
      (¬ forbidden(g)) and b1 → s1
      □
      (¬ forbidden(g)) and b2 → s2
      □
      ...
]
```

Proof: We use a well-founded induction on G. For the induction hypothesis we assume that for every $h < g$: $\vdash SFT(invariant(h), R_h)$. From this assumption, we

must prove $\vdash \mathrm{SFT}(\mathit{invariant}(g), R_g)$. To show this, we must find a well-founded set W, a parameterized invariant pi, and a helpful direction d_w for each non-minimal $w \in W$, such that the conditions of the SFT rule are valid.

Here is most of the set W:

$$\{w \in G \mid w \leq g, \text{ and } \mathit{forbidden}(w) = \mathit{forbidden}(g)\},$$

with the order for W inherited from G. For each $w \in W$ define $pi(w) = \mathit{invariant}(w)$ and $d_w = e_{level(w)}$. We also add one more element $\bullet$ to W with $pi(\bullet) = \mathit{forbidden}(g)$, and $\bullet$ below every element in W. We now need to prove the five conditions of the rule.

IOE Case 1: Prove IOE for a non-$\bullet$ element $w \in W$, with $w \neq g$. Since $w \neq g$ it follows that $w < g$ and the induction hypothesis implies: $\vdash \mathrm{SFT}(\mathit{invariant}(w), R_w)$. Since $\mathit{invariant}(w) = pi(w)$, and $\mathit{forbidden}(w) = \mathit{forbidden}(g)$, this implies $\vdash \mathrm{SFT}(pi(w), R_g)$. Let $\bar{R}$ be the repetition statement, obtained from R_g by strengthening its guards in this way:

```
*[
      (¬ b_{d_w}) and (¬ forbidden(g)) and b_1  →  s_1
      □
      (¬ b_{d_w}) and (¬ forbidden(g)) and b_2  →  s_2
      □
      ...
]
```

Since $\bar{R}$ was obtained by strengthening the guards in R_g, Lemma 3.2 implies $\vdash \mathrm{SFT}(pi(w), \bar{R})$, which is precisely what's needed for IOE to hold for w.

IOE Case 2: Prove IOE for $w = g$. That is, we must show $\vdash \mathrm{SFT}(pi(g), \bar{R})$, where $pi(g) = \mathit{invariant}(g)$ and $\bar{R}$ is obtained from R_g by strengthening it's guards (as in the previous case). Let $\mathit{child}_b(g)$ be the set of elements in G which are required by rule (b) of Definition 4.4 as immediate descendants of g, and let $\Gamma = \{\mathit{invariant}(w) \mid w \in \mathit{child}_b(g)\}$. This definition of Γ meets the requirements of Lemma 3.5 (taking p in that lemma to be $pi(g)$, and taking R in that lemma to be $\bar{R}$). Therefore Lemma 3.5 implies the required condition of IOE: $\vdash \mathrm{SFT}(pi(g), \bar{R})$.

INIT: Note that $invariant(g) = pi(g)$, and g is an element of W. Therefore, $invariant(g)$ implies that there exists some $w \in W$ such that $pi(w)$.

NOINC: Let w be a non-$\bullet$ element of W and let i be a direction. We must prove:

$$\{pi(w) \text{ and } b_i\}\ s_i\ \{\text{There exists } v \text{ such that } v \leq w \text{ and } pi(v)\}.$$

Suppose $x \in \Sigma$ is a state that satisfies the precondition of this assertion, and consider some $y \in \Sigma_\perp$ such that $(x, y) \in s_i$. The state y cannot be $\perp$ (since then it is possible to construct a finite computation path of R which starts in a state satisfying p and ends with $\cdots, x, i, \perp$, and this contradicts SFT(p, R)). If y satisfies $forbidden(w)$, then the postcondition is satisfied by taking $v = \bullet$. And if y doesn't satisfy $forbidden(w)$, then the postcondition is satisfied by taking $v = w$, since

$$invariant(w) = reachable(s, forbidden(w)).$$

In both cases we have shown that y satisfies the postcondition.

DEC: Let w be a non-$\bullet$ element of W, and let $i = level(w)$ and recall that $d_w = e_{level(w)} = e_i$. We must prove:

$$\{pi(w) \text{ and } b_{d_w}\}\ s_{d_w}\ \{\text{There exists } v \text{ such that } v < w \text{ and } pi(v)\}.$$

Suppose $x \in \Sigma$ is a state that satisfies the precondition of this assertion and consider some $y \in \Sigma_\perp$ such that $(x, y) \in s_{d_w}$. The state y cannot be $\perp$ (for the same reason as the previous paragraph). If y satisfies $forbidden(w)$, then the postcondition is satisfied by taking $v = \bullet$. And if y doesn't satisfy $forbidden(w)$, then consider rule (c) (in the construction of G). Since x satisfies the precondition of the assertion, it follows that x is also in $step(d_w, pi(w))$ which is the same as $step(e_i, invariant(w))$. Thus, rule (c) states that there is an element $v \in G$ with $v < w$ such that

$$invariant(v) = reachable(y, forbidden(v))$$

and

$$forbidden(v) = forbidden(w).$$

This v is in W and $pi(v)$ holds for y, so once again y satisfies the postcondition.

TERM: For each minimal element w of W we must show that

$$pi(w) \text{ implies } (\neg b) \text{ or } forbidden(g).$$

(Recall that b is the disjunction of all of R's guards.) The only minimal element of W is $\bullet$, and $pi(\bullet) = forbidden(g)$, and

$$forbidden(g) \quad \text{implies} \quad (\neg b) \text{ or } forbidden(g).$$

This completes the proof of Lemma 4.6. □

We now have enough to prove $\vdash SFT(p, R)$:

Lemma 4.7. $\vdash SFT(p, R)$.

Proof: Take $\Gamma = \{invariant(g) \mid g \in G \text{ and } level(g) = 0\}$. As a result of Lemma 4.6, this definition of Γ meets the conditions of Lemma 3.3, therefore: $\vdash SFT(\vee\Gamma, R)$. Moreover, $\vee\Gamma = p$, hence we have the required result. □

In this section, we assumed that $SFT(p, R)$, and from this concluded $\vdash SFT(p, R)$ (Lemma 4.7). This is summarized as the following theorem:

Theorem: (Completeness of SFT) For a repetition statement R and a predicate p:

$$\text{if } SFT(p, R), \text{ then } \vdash SFT(p, R).$$

□

5. Extreme Fairness

Pnueli [9] introduced *extreme fairness* as a method for reasoning about certain kinds of probabilistic nondeterminism, within the fairness paradigm. The typical example is a repetition statement like this:

```
*[
      x > 0  →  x := 2
      □
      x > 0  →  x := x − 1
]
```

There is an infinite strongly-fair execution path: the path alternates between the first and second directions. But consider a probabilistic selection mechanism which selects the first direction with some probability p $(0 < p < 1)$ and the second direction with probability $1-p$. Such a selection mechanism is guaranteed (with probability 1) to terminate, since with probability 1 any infinite execution path will have some selection of the first direction, followed by two consecutive selections of the second direction.

The reason that this statement has infinite strongly-fair paths is simple enough: although a strongly fair path must choose the second direction infinitely often, it need never choose the second direction when $x = 1$ holds. This is the motivation for the definition of extremely-fair execution paths and extremely-fair termination:

Definition. (Extremely-Fair Execution Path) Let R be a repetition statement with guards $b_1, b_2, \cdots$ and executable statements $s_1, s_2, \cdots$. Let Γ be a countable set of predicates. Let $x_1, d_1, x_2, d_2, \cdots$ be a (possibly infinite) execution path for R. The path is *strongly* Γ*-extremely-fair* provided that the following holds for every direction d of R and every predicate $\gamma \in \Gamma$:

> If $\{i \mid b_d$ and γ hold for $x_i\}$ is infinite, then direction d is selected infinitely often from states where b_d and γ both hold.

□

Definition. (Extremely-Fair Termination) For a repetition statement R, a countable set of predicates Γ, and a predicate p, we use the notation Γ–SFT(p,R) to denote the statement that R has no strongly Γ-extremely fair execution path that starts in a state that satisfies p and is infinite or ends with $\perp$. □

For $\Gamma = \{true, x = 1\}$, the repetition statement given above contains no infinite strongly Γ-extremely-fair execution paths. Here's why: such a path must eventually select the first direction, after which the value of x may only be 1 or 2. Also, after this point the second direction will be selected infinitely often, and since the path is

infinite, this may only occur when $x = 2$. But, after executing the second direction, the value of x will be 1, and this will occur infinitely often. Therefore, the second direction must eventually be selected when $x = 1$ holds, and after this the execution terminates.

Pnueli took the set Γ as the set of first-order definable predicates over Σ. He showed that for a finite number of directions, and this Γ, strongly Γ-extremely fair termination implies termination (with probability 1) for any probabilistic selection mechanism (independent of the actual probabilities).

The extension to arbitrary countable Γ was proposed by Francez [6, Section 4.3], who also showed the soundness of a rule similar to Γ–SFT, shown in Figure 2.

Let Γ be a countable set of predicates, let p be a predicate, and let R be a repetition statement:

```
*[
        b1 → s1
        □
        b2 → s2
        □
        ...
]
```

To prove Γ–SFT(p,R): Choose a well-founded, partially-ordered set (W,<), a predicate $pi(w)$ for each $w \in W$, a direction d_w and a predicate $\gamma_w \in \Gamma$ for each non-minimal $w \in W$, all satisfying:

1. (INIT) p implies there exists w such that $pi(w)$.
2. (TERM) For all minimal $w \in W$: $pi(w)$ implies none of R's guards hold.
3. (DEC) For all non-minimal $w \in W$:

   $\{pi(w)$ and b_{d_w} and $\gamma_w\}$ s_{d_w} {There exists v such that $v < w$ and $pi(v)\}$.
4. (NOINC) For all non-minimal $w \in W$ and all directions i:

   $\{pi(w)$ and $b_i\}$ s_i {There exists v such that $v \leq w$ and $pi(v)\}$.
5. (IOE) For all non-minimal $w \in W$, Γ–SFT($pi(w), \bar{R}$) can be proved by an application of this rule, where $\bar{R}$ is the following repetition statement:

```
*[
        (¬ (b_dw or γw)) and b1 → s1
        □
        (¬ (b_dw or γw)) and b2 → s2
        □
        ...
]
```

Figure 2. Γ–SFT Rule

We will write $\vdash \Gamma\text{–SFT}(p, R)$ to denote that $\Gamma\text{–SFT}(p, R)$ can be proved with the Γ–SFT rule of Figure 2. The question of completeness of this rule was left open by Francez [6, Page 126]. But making use of the soundness and completeness of SFT for countably infinite directions, the soundness and completeness of Γ–SFT is easy:

Theorem: (Soundness and Completeness of Γ–SFT) For a countable set of predicates Γ, a repetition statement R and a predicate p:

$$\Gamma\text{–SFT}(p, R) \text{ iff } \vdash \Gamma\text{–SFT}(p, R).$$

Proof: First, create a new repetition statement $\bar{R}$ such that for each direction "$b \rightarrow s$" in R and each $\gamma \in \Gamma$, the repetition statement $\bar{R}$ has a direction "b and $\gamma \rightarrow s$". Note that $\bar{R}$ has only countably many directions, and that:

$$\Gamma\text{–SFT}(p, R) \text{ iff } \text{SFT}(p, \bar{R})$$

and

$$\vdash \text{SFT}(p, \bar{R}) \text{ iff } \vdash \Gamma\text{–SFT}(p, R)$$

Since the SFT rule is sound and complete, we also know $\text{SFT}(p, \bar{R})$ if and only if $\vdash \text{SFT}(p, \bar{R})$. Combining this with the above two equivalences yields the needed result. □

6. Conclusion

The primary result of this paper is that the usual rule for proving strongly fair termination remains sound and complete, even when the nondeterminism has countably infinite directions. As a consequence, a slight variation of this rule is sound and complete for strongly Γ-extremely fair termination.

Acknowledgement

I wish to thank Nissim Francez for a careful reading of the first draft of this paper. He made a number of critical corrections, including the definition of SFT(p, R), and made suggestions about how to formulate $\vdash$ SFT(p, R).

References

(1) K.R. Apt and E.-R. Olderog. Proof rules and transformations dealing with fairness, *Science of Computer Programming* 3 (1983), 65-100.

(2) K.R. Apt, A. Pnueli and J. Stavi. Fair termination revisited - with delay, *Theoretical Computer Science* 33 (1984), 65-84.

(3) H.J. Boom. A weaker precondition for loops, *ACM TOPLAS* 4 (1982), 668-677.

(4) E.W. Dijkstra. Guarded commands, nondeterminacy and formal derivation of programs, *CACM* 18 (1975), 453-457.

(5) E.W. Dijkstra. *A Discipline of Programming*, (Prentice-Hall, 1976).

(6) N. Francez. *Fairness*, Springer-Verlag, New York, 1986.

(7) O. Grumberg, N. Francez, J.A. Makowsky and W.P. deRoever. A proof rule for fair termination of guarded commands, *Information and Control* 66 (1985), 83-102.

(8) D. Park. A predicate transformer for weak fair iteration. *Proc. 6th IBM Symp. on Math. Foundation of Computer Science* (1981).

(9) A. Pnueli. On the extremely fair treatment of probabilistic algorithms, in: *Proceedings of the 15th Annual ACM Symposium on Theory of Computing*, (1983), 278-290.